3 Elementary data structures

Reading: MAW Chapter 3

3.1 Lists

- A sequence of items $a_1, a_2, \ldots, a_n$.
- List operations include insert, delete, search, etc., depending on the application.
- Implementation of a list: array or linked list, which may be singly, doubly, and circular.
- The space complexity of each implementation.
- The time complexity of each operation ranges from $O(1)$ to $O(n)$, depending on the implementation.
- An example: polynomials.

  For polynomial $p(x) = 3x^3 - 2x + 1$, a list of three items is used to represent the polynomial, where each item is a pair of coefficient and degree, i.e., $(3,3), (-2,1), (1,0)$.

  Suppose a linked list is used to store the list above. How do you perform polynomial addition and multiplication?

3.2 Stacks

- A sequence of items with only one open end (the top of the stack) for insertions and deletions.
- Stack operations include push (insert to the top) and pop (delete from the top).
- Implementation of a stack: array or linked list.
- LIFO (last in first out).
- An example: balancing symbols.

  {[[(){{[]}}]{}]} is good but [(){{]}} is bad.

  Initialize a stack. Read the input one symbol at a time. If left, then push. If right, pop and compare.

3.3 Queues

- A sequence of items with one open end (the back of the queue) for insertion and the other open end (the front of the queue) for deletion.
- Queue operations include enqueue (insert to the back) and dequeue (delete from the front).
- Implementation of a queue: array (circular) or linked list.
- FIFO (first in first out) or FCFS (first come first served).
- An example: job scheduling.

  A single machine provides service to in-coming jobs. A waiting queue is maintained. When a job arrives, it is added to the end of the queue (enqueue), and when the machine becomes idle, it executes the first job in the queue (dequeue).